## *IN THE CLAIMS*

Please amend claims 1 and 2, as follows.

---

B1

--1. [Twice Amended] A multi-tasking operating system for managing simultaneous access to scarce or serially re-usable resources by multiple process threads, comprising:

at least one resource;

a plurality of threads requesting access to said resource; and

a stationary queue for allocating access to said resource amongst said threads one-by-one in order of request.--

2. [Amended] A multi-tasking operating system stationary queue for managing simultaneous access to scarce or serially re-usable resources by multiple process threads, the stationary queue comprising:

a sleep code routine for generating a unique block identifier when a process thread temporarily cannot

B1

gain access to said resource and must be suspended; and a wakeup code routine for generating a unique run identifier when a next thread in line is to be re-animated and granted access to said resource in FIFO order.

## *IN THE SPECIFICATION*

Please amend the paragraph beginning at Page 8, line 14, as follows:

B2

-- If, in step 144, it is determined that the requested resource is available, in step 145 the thread uses the resource and, when finished, in step 152 returns the resource. In step 154 it is determined whether or not NFW counter 112 equals NSC counter 110. If they are equal, there are no threads waiting on the resource, and processing returns to step 142 to await a request from a thread for the resource. If they are not equal, there is a waiting thread. In step 156, run ID 132 is created using the value of NSC 110 counter, which will then be incremented in step 158. In step 160, the thread identified by run ID 132 is awakened and made ready to run by the operating system. In step 162, that thread awakens, and in step 145, uses the resource.--